Docket No.: M4065.0459/P459

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR U.S. LETTERS PATENT

Title:

REFRESH ALGORITHM FOR MEMORIES

Inventor:

Shuba Swaminathan

Thomas J. D'Amico
DICKSTEIN SHAPIRO MORIN &
OSHINSKY LLP
2101 L Street NW
Washington, DC 20037-1526
(202) 828-2232

TITLE OF INVENTION

REFRESH ALGORITHM FOR MEMORIES

FIELD OF THE INVENTION

[0001] The present invention relates generally to memory devices, and, more particularly, to a method and apparatus for refreshing data stored in flash memory cells.

BACKGROUND OF THE INVENTION

[0002] Flash memory is a type of nonvolatile memory that can be erased in units of memory called blocks and programmed in bytes. Flash chips are divided into a plurality of main blocks. Each main block is physically isolated from each other main block. The main blocks are further divided into erase blocks. An erase block is also subdivided into sectors which contain bytes. An erase block may contain, for example, 128 sectors each comprising 512 bytes. All of the erase blocks within a main block share a common bit line.

[0003] As a result of the architecture of erase blocks, high voltages are required for programming. When a sector is programmed, the high voltages disturb the data stored in the erase blocks within the same main block. This may cause the data in the other erase blocks of the same main block to become corrupted.

[0004] Solutions to the disturb effect include both design solutions and algorithmic solutions. One such design solution is to reduce the number of erase blocks in each main block. Since only the data in erase blocks of the *same* main block is disturbed during programming, reducing the number of erase blocks in each main block reduces the number of erase blocks that are disturbed while programming an erase block. In addition, as a result of a flash chip having more main blocks, there are fewer programs executed in each main block. Both of these factors decrease the disturb effect.

[0005] However, increasing the number of main blocks also increases the resources necessary to operate the flash memory. All of the erase blocks within the same main block share certain resources (i.e. a bit line) necessary for reading and writing data. If fewer erase blocks are in each main block, these resources must be duplicated to a greater degree. As a result, the design of a flash chip becomes more complicated and less cost-effective. In addition, more of the space on the flash chip is consumed by the resources to allow the reading and writing of data. Accordingly, there is less space for data to be stored.

[0006] One algorithmic solution to the disturb effect is to alternate which erase blocks within a main block are erased and programmed. After an erase block is erased and programmed, it is not erased and programmed again until every other erase block in that main block is erased and programmed. For example, if a main block contains four erase blocks, once erase block 1 is erased and programmed, erase block 1 is not erased and programmed until erase blocks 2, 3 and 4 are erased and programmed.

BRIEF SUMMARY OF THE INVENTION

[0010] The present invention mitigates the problems associated with the prior art and provides a unique method and apparatus for refreshing data stored in a flash memory device.

[0011] In accordance with an exemplary embodiment of the present invention, a counter is maintained for each erase block of a flash memory device. When a erase block is erased, the counter for that erase block is set to a predetermined value while the counters for the other erase blocks are incremented. When a counter reaches a predetermined threshold, the data stored in the corresponding erase block is refreshed. Counters are maintained in a table with eight byte entries. Five bytes from three table entries constitute the counters for eight erase blocks. One bit from each of the fifteen bytes comprises each counter.

[0012] An algorithm that periodically refreshes data in erase blocks is desirable.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] The above and other features and advantages of the invention will be more readily understood from the following detailed description of the invention which is provided in connection with the accompanying drawings.

[0014] FIG. 1 is a processor circuit which utilizes a flash memory device constructed either in accordance with the prior art or in accordance with the present invention;

[0015] FIG. 2 is a flowchart of an exemplary embodiment of the present invention;

[0016] FIG. 3 is a flowchart of another exemplary embodiment of the present invention;

[0017] FIG. 4 is a block diagram of an entry in a table used for storing counters; and

[0018] FIG. 5 is a block diagram of a counter for an erase block.

DETAILED DESCRIPTION OF THE INVENTION

[0019] In the following detailed description, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to make and use the invention, and it is to be understood that structural changes may be made and equivalent structures substituted for those shown without departing from the spirit and scope of the present invention.

[0020] Fig. 2 shows an exemplary embodiment of the present invention. A solution to the disturb effect in which data is refreshed based on the number of times sectors are programmed is illustrated. Since it takes more than one programming operation to corrupt the data in the other erase blocks of the same main block, periodically refreshing the data obviates the problem.

[0021] Refreshing data, however, creates dead time when the flash memory device is inaccessible to the host system. Consequently, refresh algorithms have been developed to optimize the frequency with which erase blocks are refreshed. For example, flash memory device 100, as illustrated in Fig. 1, may implement the refresh algorithm illustrated in Fig. 2.

[0022] Initially, the counters for each sector in flash memory device 100 are set to a predetermined value in segment 200. This predetermined value will generally be zero. After the counters are all set to the predetermined value in segment 200, the control circuit in flash memory 100 checks if a sector was programmed in processing segment 202. If a sector was not programmed in processing segment 202, the control circuit in flash memory 100 continues checking if a sector was programmed.

[0023] If a sector was programmed, as detected in processing segment 202, the control circuit in flash memory 100 increments the programmed sector's counter in segment 204. The control circuit in flash memory 100 then determines whether any sector counters equal or exceed a predetermined threshold at processing segment 206. The predetermined threshold is set in the software. This threshold will vary for each type of flash memory device depending on many factors that effect how many times nearby sectors can be programmed without corrupting data. Alternatively, counters can be decremented until they equal or are less than a predetermined threshold.

[0024] If none of the sector counters equal or exceed the predetermined threshold, as detected at processing segment 206, the control circuit in flash memory 100 returns to processing segment 202 to determine if any sectors have been programmed.

[0025] If a sector counter equals or exceeds the predetermined threshold as detected at processing segment 206, as detected at processing segment 206, the control circuit in flash memory 100 initiates a refresh operation for the nearby sectors. The control circuit in flash memory 100 reads the information in each surrounding sector and rewrites the information in segment 208. The counter for the active sector is then set to the predetermined value in segment 210. Finally, the counters for each refreshed sector are incremented in segment 212.

[0026] This method requires maintaining counters for each sector of each block. Most flash memories have 128 sectors per erase block. Not only do the counters take up storage space that could be used for memory storage, but due to the large number of counters, the process of determining which sector counter to increment is arduous.

[0027] Fig. 3 shows another exemplary embodiment of the present invention. Fig. 3 differs from Fig. 2 in that there is only one counter per erase block instead of a counter for each sector in each erase block. This dramatically reduces the space used for counters.

[0028] In addition, instead of incrementing a counter for a sector each time a neighboring sector is written to, each time an erase block is erased, the counters for all of the erase blocks in the same main block are incremented.

[0029] Initially, the counters for all of the erase blocks in flash memory 100 are set to a predetermined value at segment 300. When an erase block is erased, as determined in processing segment 302, the control circuit in flash memory 100 resets the active erase block's counter to a predetermined value at segment 304 and increments all erase block counters respectively associated with the non-erased blocks in the same main block at segment 306. The control circuit in flash memory 100 then checks if any of the erase block counters equal or exceed a predetermined threshold at processing segment 308. The control circuit in flash memory 100 then refreshes all erase blocks with counters that equal or exceed the predetermined threshold.

[0030] The counters are stored in a table format in the flash memory. Each entry in the table, as depicted in Fig. 4, is 8-bytes. Each 8-byte entry is divided into three parts. The two most significant bytes of an 8-byte entry are always set to 'FF' (11111111) and 'F6' (11110110) to distinguish between refresh counters and other information present in the table.

[0031] The next most significant byte of an 8-byte entry is further divided into two parts. The first part, the two least significant bits, indicate whether the 8-byte entry contains the 5 most significant bits, 5 middle bits or 5 least significant bits of each of the 15 bit counters. Accordingly, these two bits can be set to 00, 01 or 10. The remaining 6 bits indicate which main block this 8-byte entry corresponds to. For example, if a flash memory has 32 erase blocks, those 32 erase blocks could be divided into 8 main blocks, each containing 4 erase blocks. A byte that contains the middle 5 counter bits for the second main block would appear as follows: 000010 01. The value in the two least significant bits is set to 01 to indicate that the byte contains the middle bits (00 = 5 least significant bits; 01 = 5 middle bits; and 10 = 5 most significant bits). The value in the 6 most significant bits is set to a value of 2 to indicate that this byte corresponds to the second main block. Accordingly, there would be two additional 8-byte entries in the table where the 6 most significant bits of the third most significant byte are set to a value of 2. One entry will have the two least significant bits of the third most significant byte set to 00, to indicate that it contains the 5 least significant bits of the counters corresponding to the second main block, and the other entry will have the two least significant bits of the third most significant byte set to 10 to indicate that it contains the 5 most significant bits of the counters corresponding to the second main block. Additionally, there will be three 8-byte table entries for each of the other erase block.

[0032] The 5 least significant bytes of each table entry contain the counters. Each of these 5 bytes contains one bit of a 15-bit counter for each of 8 erase blocks. Each counter is a 15 bit value, so it is distributed across 3 table entries 500, 502 and 504, as shown in Fig. 5. One bit from each byte corresponds to counter 506, e.g. the five least significant bits of the counter for erase block 3 506 come from table entry 500, the five middle bits of the counter for erase block 3 506 come from table entry 502 and the five most significant bits for erase block 3 506 come from table entry 504. For example, as shown in Fig. 5, the counter for erase block 3 506 is comprised of the bit 3 of each byte. Similarly, for the first erase block within a group of 8 erase blocks, the least significant bit of each byte compose the counter.

[0033] If an erase block contains user data, when that erase block is refreshed, the data is moved to a new location, that is, the data is moved and then the block is erased. By moving the data to a new location, it can be written to and read from the new location. If an erase block contains system data, such as the firmware or a BIOS, the contents refreshed "in place." The contents of the erase block are read into a temporary memory, checked for data integrity, and rewritten over the original data in its original location.

[0034] When multiple counters equal or exceed the predetermined threshold value simultaneously, flash memory device 100 may appear busy to processor 110 for an extended period of time. Since this situation is undesirable, the time spent refreshing multiple erase block can be hidden from processor 110 by allowing processor 110 to continue accessing flash memory device 100 between refresh operations. Each erase block that requires refreshing is refreshed after an operation initiated by processor 110, such as, for example, write operations. As a result, instead of using 100% of flash memory 100's capacity during multiple refreshes, and interfering with any other operations being performed by processor 110, each operation that processor 110 performs will take a little longer to complete while a single erase block is refreshed, but the flash memory device being refreshed will be accessible to processor 110 during the refreshes.

[0035] While the invention has been described with reference to exemplary embodiments various additions, deletions, substitutions, or other modifications may be made without departing from the spirit or scope of the invention. Accordingly, the invention is not to be considered as limited by the foregoing description, but is only limited by the scope of the appended claims.